

DETAILED ACTION

RESPONSE TO AMENDMENT

Claim rejections based on prior art

1. Applicant's arguments filed 02/19/2008 with respect to claims 1-26 have been fully considered but are moot in view of the new ground(s) of rejection.

The rejection(s) of claim(s) 1-26 under Derr (US 2004/0128590) in view of Ivanov et al. (US pat. 2005/0015760) have been fully considered and persuasive. However, base on the amendment, a new ground(s) of rejection is made in view of Bernasconi et al. (US pat. 6,158,018) and Ivanov et al. (US pub. 2005/0015760).

The applicant's arguments filed 02/19/2008 with respect the cited references of Bernasconi and Ivanov are not persuasive.

The applicant stated that Bernasconi and Ivanov do not disclose all the limitation in claim 1. See the rejection below for further detail.

Also, with respect to ‘modify’ a captured incoming request, see Col. 6, lines 27-33, which discloses “After the corrected DSP program software from section 20b is delivered to the DSP 16 over bus 28, the patching circuitry 22, and bus 30, a program op code causes a jump back from the end of the block of corrected DSP program software in section 20b of the RAM 20 to the remainder of good DSP program software in section 18c of the ROM 18”. See also col. 9, line 57 to col. 10, line 5, which disclose “At this stage, the patching circuitry 22 sends to the DSP 16 a branch op code followed by a branch address corresponding to the beginning of a block of corrected DSP program software in the RAM 20 such as in section 20b thereof. Therefore, in the third general phase of device operation,

the DSP 16 fetches corrected DSP program software instructions from the RAM 20 such as section 20b thereof. Moreover, corrected DSP program software can be set up to cause the DSP 16 to establish new break and branch addresses, thereby setting up additional patches to be executed during the operation of the device's DSP 16. Also, a section of corrected DSP program software typically includes, at the end thereof, jumping instructions causing the DSP 16 to recommence fetching DSP program software from the ROM 18 in a section such as 18c located downstream of the previous flawed portion 18b in the ROM 18").

The claim language doesn't specifically disclose what is a 'trigger-matching logic' or a 'control logic', and how this 'modification' takes place.

I. INFORMATION CONCERNING OATH/DECLARATION

Oath/Declaration

2. The applicant's oath/declaration has been reviewed by the examiner and is found to conform to the requirements prescribed in 37 C.F.R. 1.63.

II. INFORMATION CONCERNING DRAWINGS

Drawings

3. The applicant's drawings submitted are acceptable for examination purposes.

III. REJECTIONS BASED ON PRIOR ART

Double Patenting

The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims

Art Unit: 2181

are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

4a. **Claims 1 and 2** are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claim 16 of copending Application No. 10/781,512 in view Bernasconi et al. (US pat. 6,158,018).

4b. Initially, it should be noted that the present application and Application No. 10/781,512, share one common inventor, which is Chee Lee. The assignee for both applications is INTEL CORPORATION. The examiner also notes that neither the instant application nor U.S. application 10/781,512 were the subject of a restriction by the office.

4c. Claimed subject matter in the instant application is fully disclosed in the referenced copending application and would be covered by any patent granted on that copending application since the referenced copending application and the instant application are claiming common subject matter, as noted below. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993).

4d. Furthermore, there is no apparent reason why applicant would be prevented from presenting claims corresponding to those of the instant application in the other copending application. See MPEP § 804.

4e. Claim 1 is compared to claims 16 of application 10/781,512 in the following table:

Instant Application	Application 10/781,512
<p>(Currently Amended)</p> <p>trigger-matching logic to capture an incoming read/write request cycle from a requesting device and to determine if the captured incoming read/write request cycle matches at least one trigger condition of one or more of trigger conditions; and</p> <p>control logic coupled to the trigger-matching logic to select a single instruction upon detection of the at least one matched trigger condition and to execute one or more operations specified by the selected single instruction to modify the captured incoming read/write request cycle prior to transmission to a designated end device,</p> <p>wherein the at least one single instruction is selected based on the at least one matched trigger condition and the requesting device is different from the designated end-device (claim 1).</p>	<p>A patch module comprising:</p> <p>a trigger-matching logic to capture an incoming request cycle from a requesting device and determine if the captured incoming request cycle matches one or more of trigger conditions;</p> <p>and a control logic coupled to the trigger-matching logic to select a set of instructions upon detection of at least one matched trigger condition and to execute operations as specified by the selected set of instructions,</p> <p>wherein if the captured request cycle that caused a trigger is a non-posted request cycle, the control logic instructs a completion queue to load the completion queue with one of the following (1) unmodified header information from the captured non-posted request cycle, (2) modified header information associated with modified non-posted request cycle, or (3) header information associated with a new request cycle generated in response the captured request cycle, wherein the control logic instructs the completion queue whether or not to return a completion packet received from a designated end-device when the completion queue is loaded with header information from one of the modified non-</p>

	posted request cycle and the generated, new request cycle to the requesting device, wherein the requesting device is different from the designated end-device (claim 16).
--	---

In regards to “wherein the at one single instruction is selected based on the at least one matched trigger condition” This part of the limitation is disclosing that the control logic will select an instruction upon detection of one matched trigger condition. This limitation has already been discloses as part of the claim; for example “control logic coupled to the trigger-matching logic to select a single instruction upon detection of the at least one matched trigger condition.

This a provisional double patenting rejection since the conflicting claims have not yet been patented. The double patenting rejection is also applicable to other claim in the application; claim 2 of the instant application corresponds to claim 17 of Application 10/781,512.

Claim 16 from application number 10/781,512 didn’t specifically discloses “to modify the captured incoming request cycle prior to transmission to a designated end device,”

Bernasconi teaches “to modify the captured incoming request cycle prior to transmission to a designated end device, (See col. 6, lines 46-47, which discloses “and at that point, a branch to corrected DSP program software in section 20b is performed”. Col. 6, lines 27-33 also discloses “After the corrected DSP program software from section 20b is delivered to the DSP 16 over bus 28, the patching circuitry 22, and bus 30, a program op code causes a jump back from the end of the block of corrected DSP program software in section 20b of

the RAM 20 to the remainder of good DSP program software in section 18c of the ROM 18”)

Application number 10/781,512 and Bernasconi's invention are analogous art because they are from the same field of endeavor of a patching circuitry to bypass or fix a flaw.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to modify the patch module to process non-posted request cycles and to control completions returned to requesting device as taught by application number 10/781,512 and an improved IC device including patching circuitry to bypass flawed data stored in an embedded ROM and a method of operation therefore as taught by Bernasconi

The motivation for doing so would have been because Bernasconi teaches that **[“Moreover, the invention here permits as many corrections to the DSP program software as there are flaws in the IC's ROM 18, and quite importantly, these corrections to the software can be implemented in a final production device 12 with internal flaws in its embedded ROM 18, thereby obviating the need to design and manufacture a new chip” (col. 9, lines 43-48)].**

Therefore, it would have been obvious to combine Bernasconi and Application number 10/781,512 for the benefit of creating a patch mechanism used to detect and workaround defects and conditions existing in an integrated circuit chip to obtain the invention as specified in claim 1.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. **Claims 1-19, and 21-26**, are rejected under 35 U.S.C. 103(a) as being unpatentable over Bernasconi et al. (US pat. 6,158,018) in view of Ivanov et al. (US pat. 2005/0015760).

7. As per **claim 1**, Bernasconi discloses a system comprising:

trigger-matching logic (**the comparator 42 of the patching circuitry 22, as discloses in col. 7, lines 16-19) (col. 9, lines 51-57, teaches that the comparator is a logic that does the matching between the current DSP program address and a break address correspond to a flawed DSP program stored in the ROM 18, such as at section 18b) to capture an incoming read/write request cycle (the DSP program address, as discloses in col. 9, line 54) from a requesting device (directionally, fig. 1 shows the DSP program address coming from the DSP 16) and to determine if the captured incoming read/write request cycle matches at least one trigger condition of one or more of trigger conditions (see col. 9, lines 51-57); and**

control logic (**the patch control 60 of the patching circuitry 22, as discloses in col. 7, lines 44 and 45. See also col. 8, lines 4-7) coupled (see fig. 3, which shows the comparator 42, the trigger-matching logic, being coupled to the patch control 60) to the trigger-matching logic to select a single instruction upon detection of at least one matched trigger condition and to execute one or more operations specified by the selected single instruction to modify the captured incoming read/write request cycle prior to transmission to a designated end device (directionally, fig. 1 shows the DSP program address going to the DSP 16. Col. 8, lines 3-11 discloses “When the patch control module 60 is enabled over line 64, and the comparator 42 detects a match, the signal indicating a match over line 62 causes the patch control**

module 60 to initiate a patch sequence. Initiation of a patch sequence will render appropriate signals over the patch control outputs labelled patch.cycle 1 on line 74 and patch.cycle 2 on line 76. Details of patch sequence initiation will be discussed later with respect to device operation". This 'Operation', as sated in col. 9, line 57 to col. 10, line 5, discloses "At this stage, the patching circuitry 22 sends to the DSP 16 a branch op code followed by a branch address corresponding to the beginning of a block of corrected DSP program software in the RAM 20 such as in section 20b thereof. Therefore, in the third general phase of device operation, the DSP 16 fetches corrected DSP program software instructions from the RAM 20 such as section 20b thereof. Moreover, corrected DSP program software can be set up to cause the DSP 16 to establish new break and branch addresses, thereby setting up additional patches to be executed during the operation of the device's DSP 16. Also, a section of corrected DSP program software typically includes, at the end thereof, jumping instructions causing the DSP 16 to recommence fetching DSP program software from the ROM 18 in a section such as 18c located downstream of the previous flawed portion 18b in the ROM 18", wherein the at least one single instruction is selected based on the at least one matched trigger condition" (see col. 9, line 54 to col. 10, line 5).

but fails to disclose expressly wherein the requesting device is different from the designated end-device.

Ivanov discloses wherein the requesting device (Client 108 of fig. 2, as discloses in para. 0033) is different from the designated end-device (Distribution 102 of fig. 2) (see para. 0033, which discloses "If the bit pattern of the binary signature is found in a binary file

208, scan-patch module 204 operates to fix the security vulnerability in the binary file 208 by installing a corresponding security patch on client computer 108”).

Bernasconi et al. (US pat. 6,158,018) and Ivanov et al. (US pat. 2005/0015760) are analogous art because they are from the same field of data patching.

at the time of the invention it would have been obvious to a person of ordinary skill in the art to modify the patch mechanism used to detect and workaround defects and conditions existing in an integrated circuit chip as described by Bernasconi and a systems and methods that enable patching of security vulnerabilities in binary files as taught by Ivanov.

The motivation for doing so would have been because Ivanov teaches that (“**The detection and patching of vulnerable binary files is automatic, reliable, regression free, and comprehensive across networks on an unlimited scale. These advantages can be realized in various ways including, for example, by leveraging current anti-virus infrastructure that is widely deployed across the Internet. A divergence of security patches away from conventional service packs provides for the possibility of production of regression-free fixes for security vulnerabilities in binary files**”; see para. 0020).

Therefore, it would have been obvious to combine Ivanov et al. (US pat. 2005/0015760) with Bernasconi et al. (US pat. 6,158,018) for the benefit of creating the system to obtain the invention as specified in claim 1.

8. As per **claim 2**, the combination of Bernasconi and Ivanov discloses “the system of claim 1,”[see rejection to claim 1 above], Bernasconi further discloses wherein the trigger-matching logic and the control logic are incorporated within an Input/Output (I/O) chip (**with respect to**

this limitation, Bernasconi discloses the logic that does the matching between the current DSP program address and a break address take place inside the patching circuitry 22 that is disclosed in fig. 2. The patching circuitry 22 is shown in fig. 1 coupled to a control logic, incorporated within an Input/output (I/O) integrated circuit chip” (see fig. 1).

9. As per **claim 3**, the combination of Bernasconi and Ivanov discloses “the system of claim 1,”[see rejection to claim 1 above], Bernasconi further discloses wherein the control logic can execute an operation which involves logically combining a selected operand entry with a selected register containing information from the captured cycle (see col. 3, lines 32-45, also, in fig. 3, Bernasconi also discloses the selected captured cycle’s information is stored within a register that is now shown, as disclosed by Bernasconi in col. 8, lines 18-26).

10. As per **claims 4, 10, and 22**, the combination of Bernasconi and Ivanov discloses “the system of claim 2,”[see rejection to claim 2 above], Bernasconi further discloses wherein the control logic can execute an operation which causes a new read/write request cycle to be created and forwarded to a downstream bus of the I/O controller (see col. 3, lines 47-64).

11. As per **claim 5**, the combination of Bernasconi and Ivanov discloses “the system of claim 1,”[see rejection to claim 1 above], Bernasconi further discloses wherein the control logic can execute an operation which involves modifying the captured incoming read/write request cycle (with respect to this limitation, Bernasconi discloses a method comprising the steps of supplying the corrected software to the embedded DSP after the step of providing a branch op code followed by a branch address to the embedded DSP, and supplying data to the

embedded DSP from a portion of the embedded ROM located downstream of the flawed portion thereof after the step of supplying corrected software to the embedded DSP (see col. 3, lines 32-53). Therefore, the current DSP program address is corrected from old to new. When something is corrected, it also modified. Therefore, this limitation is inherently met).

12. As per **claims 6, 14, and 26**, the combination of Bernasconi and Ivanov discloses “the system of claim 1,”[see rejection to claim 1 above], Bernasconi further discloses wherein the control logic can execute an operation which causes a timed delay or a conditional delay to be inserted **(in col. 13, lines 56-58, Bernasconi discloses a sequencer discloses in the path module, which is known in the art to “sorts data or programs into a predetermined sequence”. Bernasconi discloses this sequencer to create a delay, as disclosed).**

13. As per **claims 7, 15, and 21**, Bernasconi discloses, capturing an incoming read/write request cycle **(the DSP program address, as discloses in col. 9, line 54)** from a requesting device **(directionally, fig. 1 shows the DSP program address coming from the DSP 16);** loading information from the captured read/write request cycle into a first register **(see fig. 3 and col. 7, lines 50-55);** comparing the information stored in the first register with one or more trigger conditions **(see fig. 3, which discloses the comparator 42 of the patching circuitry 22, as discloses in col. 7, lines 16-19. Col. 9, lines 51-57, teaches that the trigger-matching logic is the logic that doest the matching between the current DSP program address and a break address correspond to a flawed DSP program stored in the ROM 18, such as at section 18b);** selecting a single instruction based on a matched trigger condition, and executing the

selected single instruction (see col. 9, line 57 to col. 10, line 5) to modify the captured incoming read/write request cycle prior to transmission to a designated end-device (directionally, fig. 1 shows the DSP program address going to the DSP 16. Col. 8, lines 3-11 discloses “When the patch control module 60 is enabled over line 64, and the comparator 42 detects a match, the signal indicating a match over line 62 causes the patch control module 60 to initiate a patch sequence. Initiation of a patch sequence will render appropriate signals over the patch control outputs labelled patch.cycle 1 on line 74 and patch.cycle 2 on line 76. Details of patch sequence initiation will be discussed later with respect to device operation”. This ‘Operation’, as sated in col. 9, line 57 to col. 10, line 5, discloses “At this stage, the patching circuitry 22 sends to the DSP 16 a branch op code followed by a branch address corresponding to the beginning of a block of corrected DSP program software in the RAM 20 such as in section 20b thereof. Therefore, in the third general phase of device operation, the DSP 16 fetches corrected DSP program software instructions from the RAM 20 such as section 20b thereof. Moreover, corrected DSP program software can be set up to cause the DSP 16 to establish new break and branch addresses, thereby setting up additional patches to be executed during the operation of the device's DSP 16. Also, a section of corrected DSP program software typically includes, at the end thereof, jumping instructions causing the DSP 16 to recommence fetching DSP program software from the ROM 18 in a section such as 18c located downstream of the previous flawed portion 18b in the ROM 18”),

but fails to disclose expressly wherein the requesting device is different from the designated end-device.

Ivanov discloses wherein the requesting device (**Client 108 of fig. 2, as discloses in para. 0033**) is different from the designated end-device (**Distribution 102 of fig. 2**) (see **para. 0033, which discloses “If the bit pattern of the binary signature is found in a binary file 208, scan-patch module 204 operates to fix the security vulnerability in the binary file 208 by installing a corresponding security patch on client computer 108”**).

Bernasconi et al. (US pat. 6,158,018) and Ivanov et al. (US pat. 2005/0015760) are analogous art because they are from the same field of data patching.

at the time of the invention it would have been obvious to a person of ordinary skill in the art to modify the patch mechanism used to detect and workaround defects and conditions existing in an integrated circuit chip as described by Bernasconi and a systems and methods that enable patching of security vulnerabilities in binary files as taught by Ivanov.

The motivation for doing so would have been because Ivanov teaches that (**“The detection and patching of vulnerable binary files is automatic, reliable, regression free, and comprehensive across networks on an unlimited scale. These advantages can be realized in various ways including, for example, by leveraging current anti-virus infrastructure that is widely deployed across the Internet. A divergence of security patches away from conventional service packs provides for the possibility of production of regression-free fixes for security vulnerabilities in binary files”**; see **para. 0020**).

Therefore, it would have been obvious to combine Ivanov et al. (US pat. 2005/0015760) with Bernasconi et al. (US pat. 6,158,018) for the benefit of creating the method to obtain the invention as specified in claims 7, 15, and 21.

14. As per **claim 8**, the combination of Bernasconi and Ivanov discloses “the method of claim 7,”[**see rejection to claim 7 above**], Bernasconi further discloses the incoming read/write request cycle is received within an I/O controller chip (**see fig. 1 and col. 4, lines 44-48**).

15. As per **claim 9**, the combination of Bernasconi and Ivanov discloses “the method of claim 7,”[**see rejection to claim 7 above**], Bernasconi further discloses executing of the instructions comprises: logically combining a selected operand entry with a selected register containing information captured from the received read/write request cycle (**see fig. 3, and col. 3, lines 29-32**).

16. As per **claims 11-13 and 23-25**, the combination of Bernasconi and Ivanov discloses “the method of claim 7,”[**see rejection to claim 7 above**], Bernasconi further discloses wherein executing the instructions comprises: modifying a cycle type section of the incoming read/write request cycle, an address section of the incoming cycle, and a data section of the incoming read/write request cycle (**with respect to this limitation, Bernasconi discloses a method comprising the steps of supplying the corrected software to the embedded DSP after the step of providing a branch op code followed by a branch address to the embedded DSP, and supplying data to the embedded DSP from a portion of the embedded ROM located downstream of the flawed portion thereof after the step of supplying corrected software to the embedded DSP (see col. 3, lines 32-53). Therefore, the current DSP program address is corrected from old to new. When something is corrected, it also modified. Therefore, this limitation is inherently met. Also, a cycle type section of the incoming cycle consists an address section of the incoming cycle, which is a form of data**).

17. As per **claim 16**, the combination of Bernasconi and Ivanov discloses “the patch module of claim 15,”[see rejection to claim 15 above], Bernasconi further discloses wherein the patch module (**the patching circuitry 22, fig. 1**) is embedded within an I/O controller chip (**see fig. 1 and col. 4, lines 44-48**) and can be programmed by a user to workaround conditions and defects existing in the I/O controller chip (**see col. 10, lines 55-60**).

18. As per **claim 17**, the combination of Bernasconi and Ivanov discloses “the patch module of claim 15,”[see rejection to claim 15 above], Bernasconi further discloses wherein the instruction execution unit can execute an instruction that comprises: a first field to specify a type of operation to be performed (**col. 2, line 63 to col. 3 line 17**), wherein the type of operations identified by the first field includes (1) timed delay operation, (2) conditional delay operation, (3) generating new cycle operation, and (4) modifying the capture request cycle operation (**see col. 2, line 63 to col. 3 line 17**); and a second field to specify whether or not a cycle generated by the instruction is to be forwarded to downstream bus (**see col. 3 lines 32-54**).

19. As per **claim 18**, the combination of Bernasconi and Ivanov discloses “the patch module of claim 15,”[see rejection to claim 15 above], Bernasconi further discloses a third field to select a register to modify (**col. 12, lines 43-47**); a fourth field to select an operand entry from an operand array (**col. 12, lines 44-55**); and a fifth field to select a logic gate for combining the selected register with the selected operand entry (**col. 12, lines 47-64**).

20. As per **claim 19**, the combination of Bernasconi and Ivanov discloses “the patch module of claim 17,”[see rejection to claim 17 above], Bernasconi further discloses wherein the

captured incoming cycle is a non-posted cycle (**see request 24 in fig. 1, which is an incoming request that has not been process or complete. The applicant discloses that “non-posted cycle” is request that require completion**).

21. **Claim 20** is rejected under 35 U.S.C. 103(a) as being unpatentable over Bernasconi et al. (US pat. 6,158,018) in view of Ivanov et al. (US pat. 2005/0015760), as applied to claim 19, and further in view of Hagan et al. (US pat. 5,966,547).

22. As per **claim 20**, the combination of Bernasconi and Ivanov discloses “the patch module of claim 17,”[**see rejection to claim 17 above**], Bernasconi further discloses “wherein the instruction execution unit (**the patch control 60 of the patching circuitry 22, as discloses in col. 7, lines 44 and 45- see also col. 8, lines 4-7**) can execute an instruction that comprises; a fifth field (**a fifth step**) to specify whether to load unmodified header information (**the fault address of the DSP program**) from the captured non-posted cycle (**faulty DSP program**) or loaded with modified header information (**the corrected address of the DSP program**) associated with modified request cycle (**corrected DSP program**) that is generated by the control logic (**see col. 9, lines 51-57**); and a sixth field to specify whether or not a completion associated with the capture request cycle is to be discarded (**see col. 9, line 57 to col. 10, line 5**).

Bernasconi’s and Ivanov fail to specifically discloses a RAM or ROM wherein a completion queue is to be loaded with modified header information associated with modified request cycle that is generated by the control logic”

Hagan discloses “a RAM or ROM wherein a completion queue is to be loaded with

modified header information associated with modified request cycle that is generated by the control logic” (see abst. and col. 4, lines 47-59).

Bernasconi et al. (US pat. 6,158,018), Ivanov et al. (US pat. 2005/0015760) and Hagan et al. (US pat. 5,966,547) are analogous art because they are from the same field of endeavor storing data into queue from a processor.

In view of such teaching, at the time of the invention it would have been obvious to a person of ordinary skill in the art to modify the patch mechanism used to detect and workaround defects and conditions existing in an integrated circuit chip as described by Bernasconi, a systems and methods that enable patching of security vulnerabilities in binary files as taught by Ivanov , and a method and apparatus for efficiently posting entries to a shared queue within the data processing system as taught by Hagan.

The motivation for doing so would have been because Hagan teaches that [**“The significant instructions generally translate into increased processing time, slowing the response of the processor determining whether the current queue entry is empty. Therefore, it would be advantageous to have an improvement with an apparatus for reducing the processing overhead for multiple processor or embedded processor architectures in posting events or tasks to a queue”**(col. 1, lines 49-56)].

Therefore, it would have been obvious to combine Hagan with Bernasconi and Ivanov for the benefit of creating a patch mechanism used to detect and workaround defects and conditions existing in an integrated circuit chip to obtain the invention as specified in claim 20.

IV. RELEVANT ART CITED BY THE EXAMINER

23. The following prior art made of record and not relied upon is cited to establish the level of skill in the applicant's art and those arts considered reasonably pertinent to applicant's disclosure. See MPEP 707.05(c).

24. The following references teach a computer system used to detect, transfer data, workaround defects and conditions existing in an integrated circuit chip.

U.S. PATENT NUMBER

US 6,891,765; 6,925,521; 7,243,206

V. CLOSING COMMENTS

Conclusion

a. STATUS OF CLAIMS IN THE APPLICATION

25. The following is a summary of the treatment and status of all claims in the application as recommended by **M.P.E.P. 707.07(i)**:

a(1) CLAIMS REJECTED IN THE APPLICATION

26. Per the instant office action, claims 1-26 have received a final action on the merits.

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on

the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

b. DIRECTION OF FUTURE CORRESPONDENCES

27. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ernest Unelus whose telephone number is (571) 272-8596. The examiner can normally be reached on Monday to Friday 9:00 AM to 5:00 PM.

IMPORTANT NOTE

28. If attempts to reach the above noted Examiner by telephone is unsuccessful, the Examiner's supervisor, Mr. Alford Kindred , can be reached at the following telephone number: Area Code (571) 272-4037.

The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Application/Control Number: 10/781,371
Art Unit: 2181

Page 21

Examiner
Art Unit 2181

/E. U./

Examiner, Art Unit 2181

/Alford W. Kindred/

Supervisory Patent Examiner, Art Unit 2181